Meet Crosswalk
New HTML5 Runtime

Sakari Poussa
Intel
Outline

• What is Crosswalk and why do we need it?
• Architecture – how Crosswalk is constructed?
• Features for Tizen 3.0
• How to Contribute
• Demo
What and Why
What is Crosswalk

• New HTML5 runtime based on Blink and Content Module
• Designed for Tizen but also for Android
  • Supports also Linux, Mac and Windows
• The HTML5 runtime for Tizen 3.0
  • Replaces the WebKit based Tizen 2.x WRT
• Open source - started in Sep-2013
• 6 weeks release cadence. Stable, Beta and Canary channels
• GitHub for code and reviews. JIRA for features and bugs. FreeNode for IRC.
Crosswalk Project Goals

- Fully open source project – embraces participation
- Based on W3C standards and landing zone for new draft APIs
- Bring web applications to the next level – closer to native
- Backwards compatible with Tizen 2.x WRT
- Easy adaptation for downstream projects
  - Tizen, Tizen SDK, Cordova, Intel XDK
- Good co-operation with upstream projects
  - Chromium, Blink, Skia, V8, Wayland
Why do we need new HTML5 Runtime

- Fear of WebKit project not meeting the Tizen needs
- During 2013 Blink rendering engine became live
- We believe Blink is the most competitive HTML5 engine
- Lot of other companies and communities has made the same conclusion and moved using Blink
- Google is very open and willing to accept contributions to Blink
Architecture for Tizen 3.0
Crosswalk Architecture Goals

- Based on Blink and selected parts of Chromium
- Work on the upstream to enable features we need
- Minimize the changes on Crosswalk Blink and Chromium
- API extensions in separate repositories
  - Tizen Device APIs, Cordova APIs, early or experimental W3C APIs
Crosswalk Architecture - Modules

**Crosswalk**
- Runtime model
- Manifest Parsing
- JS Extension framework
- SysApps APIs*

**Chromium**
- Cross-platform libraries (base, ipc)
- Content module: Content rendering/interaction using sandboxed processes
- Aura framework: Windowing and compositing
- Ozone
- Compositor

**GPU Command Buffer**

---

**Blink Web Engine**

**Network stack**
- Optimized for web use cases, often known as libnet

**FFmpeg**
- Audio/Video codec library

**Skia**
- 2D drawing, hardware accelerated for 2D canvas

**VB**
- Modern JavaScript engine

**WebRTC**
- Real-time communication infrastructure

---

**External Extensions**
- Tizen APIs
- Custom Native APIs

---

* The SysApps API may be part of Blink/Chromium in the future.
Crosswalk Architecture - Runtime

- Shared process model
- BP is shared with all WebApps
- WebApp contains EP and RP
- RP is sandboxed and can’t do OS calls
- RP delegates OS calls to BP via IPC
- EP is not sandboxed and can do OS calls
Features for Tizen 3.0
New features and APIs

- **Web Components** (http://www.w3.org/TR/components-intro/)
  - Future of the web app design
- **Service Worker** (http://www.w3.org/TR/service-workers/)
  - Closing the gap between the native and web applications
- **Responsive Design**
  - Media queries (L4), @viewport (http://dev.w3.org/csswg/css-device-adapt/)
  - Picture element, srcset attribute
- **Native Client**
  - Portable version, pNaCl
- **Manifest** (http://w3c.github.io/manifest/)
  - Standard manifest for web applications
- **W3C SysApps: Raw Sockets** (http://www.w3.org/2012/sysapps/tcp-udp-sockets/)
- **W3C SysApps: Device Capabilities** (http://www.w3.org/2012/sysapps/device-capabilities/)
- **W3C SysApps: App URI** (http://www.w3.org/2012/sysapps/app-uri/)
New features and APIs (cont.)

- W3C Promises API
- W3C Resource Timing API (http://www.w3.org/TR/resource-timing/)
- W3C User Timing API (http://www.w3.org/TR/user-timing/)
- W3C Ambient Light API
- W3C GamePad API
- W3C NFC API
- EcmaScript SIMD
- W3C WebRTC
- W3C Web Animations
- HTML5 input enhancements
  - context menu, pattern attribute, data list element, autocomplete
- Beacon (http://www.w3.org/TR/beacon/)
- Vehicle API (IVI)
- DLNA API (IVI)
Existing 2.x Web Features are supported

All the 2.x major features supported including
- Tizen Device APIs
- Security model and API permissions
- WebView for EFL applications
- W3C Widgets including Tizen extensions
- Cordova 3.x APIs
- W3C APIs (latest versions). Notable updates below.
  - CSS selectors (level 1&2)
  - Touch Events
  - SVG
  - Server Sent Events
  - Indexed DB
  - CORS
  - Drag and Drop
  - Web Notifications
Security

- **Tizen 3.0 new security model**
  - Compact 3-domain Smack policy for access control
  - Cynara policy checker service for API permission control
- **Crosswalk will be supporting both (Smack and Cynara)**
- **API permission checks are for**
  - All Tizen Device APIs
  - Experimental W3C APIs (e.g. SysApps APIs)
  - W3C Geolocation, getUserMedia, FullScreen, Web Notifications and Storage APIs (WebSQL, IndexDB, and FileSystem)
Crosswalk and Cynara

- Policy is created during WebApp installation
- API permission is checked against the policy during runtime
- Policy contains <application context>, <privilege> tuple
- Permission check has simple answer: ALLOW, DENY or ASK USER
How to Contribute for Crosswalk
Source Code and Build

• **Upstream is in GitHub**
  • [https://github.com/crosswalk-project](https://github.com/crosswalk-project)
  • Multiple repositories – for Tizen the relevant are
    • crosswalk
    • tizen-extensions-crosswalk

• **Tizen.org is updated daily from the upstream**
  • platform/framework/web/crosswalk.git
  • platform/framework/web/tizen-extensions-crosswalk.git

• **Build and Install**

  $ git clone ssh://poussa@review.tizen.org:/platform/framework/web/crosswalk.git
  $ gbs build -A x86_64 # RPM is ready for device installation
  $ rpm -ivh crosswalk-7.35.139.0-0.x86_64.rpm # On the device.
Demo
Demo – W3C NFC API and Sample App

- Web Components
- Polymer
- Crosswalk
- Tizen Extension
- Promises
- W3C NFC API
- NFC tag
- Tizen IVI on NUC
- Android Phone

http://www.w3.org/TR/nfc/